UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/746,978	12/21/2000	Pascal L. Renard	SC91211A	8364

7590 10/28/2003

Motorola, Inc
Austin Intellectual Property Law Section
7700 West Parmer Lane
MD: TX32/PL02
Austin, TX 78729

EXAMINER	
VO, TED T	

ART UNIT	PAPER NUMBER
2122	4

DATE MAILED: 10/28/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/746,978	Applicant(s) RENARD ET AL.	
	Examiner Ted T. Vo	Art Unit 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 December 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. This action is in response to the application filed on 12/21/2000.

Claims 1-19 are original claims.

Claims 1-19 are pending in the application.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title.

The claims 14-19 are rejected under 35 U.S.C 101 because the claimed invention is directed to non-statutory subject matter.

As per claims 14-15:

The claims 14-15 are rejected under 35 U.S.C 101 because the claimed invention is directed to non-statutory subject matter.

Despite the invention is in technological art, and claims 14-15 are claiming an algorithm being executed by a processor, claims 14-15 lack further limitations for producing a result limited to practical applications.

Analysis:

Per Claim 14: "executing a first loop including a first instruction within a body of the first loop and a second loop including a first instruction within a body of the second loop, wherein the first instruction of the first loop and the first instruction of the second loop are a same instruction at the same address".

Claim 14 is merely an execution of **a loop in which the looping functionality is not realized.** The claim language is solely an execution of a program per se. Loop body exists in every computer program, and looping **without functionality to be realized** does not produce a concrete and tangible result.

Per claim 15: "executing a first loop including a last instruction within a body of the first loop and a second loop including a last instruction within a body of the second loop, wherein the last instruction of the first loop and the last instruction of the second loop are the same instruction at the same address".
Claim 15 is merely an execution of **a loop in which the looping functionality is not realized.**

According to analysis: Even though claims 14-15 are recited as a process because of being executed by a processor, not all processes are statutory under 35 U.S.C. 101. (Schrader, 22 F.3d at 296, 30 USPQ2d at 1460). A loop is a common body of a program. If it lacks functionality to be realized or producing a transformation to lead a practical application, will be solely a program per se.

MPEP 2100: DETERMINE WHETHER THE CLAIMED INVENTION COMPLIES, WITH 35 U.S.C. 101, section IV(B)(2)(b) determines that a process claim would not be statutory under 35 USC 101 if it fails to produce a concrete and tangible result.
Despite claims 14-15 recite looping associated with a processor execution, **without functionality to be realized,** the claims are merely a program per se. Therefore, the claims merely manipulate abstract ideas in general without limitation to a practical application. Claims 14-15 are held nonstatutory and rejected under 35 U.S.C. 101.

As per claims 16-19:

The claims 16-19 are rejected under 35 U.S.C 101 because the claimed invention is directed to non-statutory subject matter.
Despite the invention is in technological art, claims 16-19 are merely a data structure per se that is not limited to practical applications.
Analysis:
Claims 16-19 are led by independent claim 16 in which it claims, "A processor instruction comprising: a first field that indicates a first termination condition for a first execution loop; and a second field that indicates a second termination condition for a second execution loop".

The claimed limitation is expressed as fields in a processor instruction. The fields without functionality to be realized hold the claim to be a data structure per se. The dependent claims 17-18 of claim 16 further include more fields in the processor instruction.

According to the analysis, without further limitation functioned for performing a practical application, the claims are merely a data structure per se.

Therefore, the claims merely manipulate abstract ideas in general without limitation to a practical application. Claims 16-19 are held nonstatutory and rejected under 35 U.S.C. 101.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

(a) Claims 1-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stoodley, "Software Pipelining Loops With Conditional Branches", 1996 IEEE.

As per claim 1:

Stoodley discloses an architecture (processor) (page 269, figure 6) and techniques for generating code for loops for VLIW architecture. The architecture executes VLIW instructions (see page 270, first column, second paragraph) included with looping branches.

The teaching of Stoodley covers claim limitation:

"*A processor comprising: an execution unit* (page 269, figure 6) *to execute a set of instructions (VLIW) an instruction fetching mechanism that retrieves the set of instructions to be executed by the execution unit*

(figure 6), at least one of the set of instructions comprising a single instruction (see page 263, first column, second paragraph, 'If the execution...') *that provides for execution of other instructions of the set of instructions* (see page 263, first column, second paragraph, 'If the execution...', particularly, True path, False path) *in accordance with multiple looping constructs*".

Stoodley's disclosure particularly addresses single looping construct that makes true path and false path (figure 1 block A).

Stoodley lacks teaching "multiple looping constructs". However, Stoodley suggests looping of more than one condition branch resulting in more than two iteration paths (loops) (see page 263, first column, second paragraph, 'If the execution...', particularly, see resulting in more than two iteration paths), and discusses difficulties of creating VLIW instructions containing operations from different iterations (Multiple looping constructs) (see page 265, second column, third paragraph, 'The second difficulty...').

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to modify an instruction to perform multiple iteration paths (multiple loops) in a suitable architecture for dealing with more than two iteration paths for covering complex loop structures usually existed in a real computer programming algorithm for improving loop performance.

As per claim 2: Regarding, "The processor of claim 1, wherein the single instruction initializes a plurality of loops for later execution", Stoodley shows a single instruction with a condition branch for performing at least two iteration paths (figure 1, block A).

Official notice is taken that "initializes a plurality of loops for later execution" is well known feature in the art. Exception handling is known for modifying a branch target to a target routine. It causes late execution of the portion of code in a program that is affected by branching.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to include branch modifying that will cause initializing a plurality of loops for later execution for conforming to exception handling.

As per claim 3: Regarding, "*The processor of claim 1, wherein the multiple looping constructs are implemented in a nested structure*", the claim is inherent in a loop structure of a given program (see page

263, first column, second paragraph, 'If the execution...', particularly, see resulting in more than two iteration paths).

As per claim 4: Regarding, "*The processor of claim 1, wherein the single instruction includes a plurality of loop termination conditions*", Stoodley discloses a single loop termination condition, for example 'bneqz.s' in block A of figure 1. However, Stoodley suggests 'resulting in more than two iteration paths". Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to modify multiple terminations for conforming to the requirement of multiple branching.

As per claim 5: Regarding, "*The processor of claim 1, wherein the single instruction includes at least one field that identifies a location of a last instruction of a loop*", Stoodley teaches the identification inherently in branch directions using the flow diagram of figure 1.

As per claim 6: Claim 6 is a method that follows an algorithm in accordance to functionality of claim 1. Claim 6 is rejected in the same reason as set forth in connection to the rejection of claim 1.

As per claim 7: where the claimed recitation, "an end-of-loop indicator, a start of loop indicator, a loop count, a loop register, and a condition code selection", is inherent in nested structure in accordance to the functionality of claim 3. Claim 7 is rejected in the same reason as set forth in connecting to the rejection of claim 3.

As per claim 8: Claim 8 has the functionality corresponding to the functionality of claim 5. Claim 8 is rejected in the same reason as set forth in connecting to the rejection of claim 5.

As per claim 9: Claim 9 has the functionality corresponding to the functionality of claim 1; Claim 9 is rejected in the same reason as set forth in connecting to the rejection of claim 1.

As per claim 10: Regarding, "*A method of executing at least one instruction by a processor, the method comprising: fetching a first instruction from a memory source (see figure 6, 'From code memory'); decoding the first instruction to identify an instruction type (see figure 6, 'Instruction decoder'); and determining a loop type selected* (see figure 6, 'Operations to Execute') *from one of a conditional and nonconditional type of loop termination for a loop that contains more than instruction other than the first instruction*",

Stoodley addresses a single looping construct that makes True path and False path (figure 1 block A).

Stoodley lacks teaching "*from one of a conditional and nonconditional type of loop termination for a loop*". However, Stoodley suggests looping of more than one condition branch resulting in more than two iteration paths (see page 263, first column, second paragraph, 'If the execution...', particularly, see resulting in more than two iteration paths) and discusses difficulties of creating VLIW instructions containing operations from different iterations (see page 265, second column, third paragraph, 'The second difficulty...').

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to modify an instruction to perform multiple iteration paths with conditional or non conditional type in a suitable architecture for dealing with more than two iteration paths for covering complex loop structures usually existed in a real computer programming algorithm for improving loop performance.

As per claim 11: Regarding, "*A method of executing at least one instruction by a processor, the method comprising: fetching a first instruction from a memory source (see figure 6, 'From code memory'); decoding the first instruction to identify an instruction type (see figure 6, 'Instruction decoder'); and determining a loop type selected (see figure 6, 'Operations to Execute') from one of a conditional and nonconditional type of loop termination for a loop that contains at least one instruction that may be interrupted during loop execution*",

Stoodley addresses a single looping construct that makes True path and False path (figure 1 block A).

Stoodley lacks teaching "*from one of a conditional and nonconditional type of loop termination for a loop*". However, Stoodley suggests looping of more than one condition branch resulting in more than two iteration paths (see page 263, first column, second paragraph, 'If the execution...', particularly, see resulting in more than two iteration paths) and discusses difficulties of creating VLIW instructions

containing operations from different iterations (see page 265, second column, third paragraph, 'The second difficulty...').

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to modify an instruction to perform multiple iteration paths with conditional or non conditional type in a suitable architecture for dealing with more than two iteration paths for covering complex loop structures usually existed in a real computer programming algorithm for improving loop performance.

Regarding, "*a loop that contains at least one instruction that may be interrupted during loop execution*", Stoodley shows a single instruction with a condition branch for performing at least two iteration paths (figure 1, block A).

Official notice is taken that an instruction that may be interrupted during loop execution is well known in the art. Interrupting is occurred by errors or exception handling that is often at branch instructions.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to include branch modifying. This will cause interrupt for conforming to exception handling or error injection.

(b) Claims 12-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stoodley, "Software Pipelining Loops With Conditional Branches", in further view of Albert et al., "Data Parallel Computers and the FORALL Statements" 1990 IEEE.

As per claim 12: Regarding, "A method of executing at least one instruction by a processor, the method comprising: fetching a first instruction from a memory source; decoding the first instruction to identify an instruction type; performing a first set of logic relating to a first loop termination condition for a first loop; and performing a second set of logic relating to a second loop termination condition for a second loop",

Stoodley discloses an execution mechanism (Stoodley: see, figure 6) for fetching instruction from memory source (Stoodley: see figure 6, 'From code memory'). The fetching is included with a decoder for

identifying a type of execution (operation that including looping, Stoodley: figure 1). Stoodley discloses one loop mechanism provided with iteration paths (True or False).

Stoodley lacks disclosing two-loop mechanism that provides more than one termination condition. However, Stoodley suggests looping of more than one condition branch resulting in more than two iteration paths (see page 263, first column, second paragraph, 'If the execution...', particularly, see resulting in more than two iteration paths) and discusses difficulties of creating VLIW instruction containing operations from different iterations (see page 265, second column, third paragraph, 'The second difficulty...').

Alberts discloses a data structure (Albert: see page 392, first column, lines 1-3: statement: FORALL (I=1:10, J=1:10)). This data structure when performed by a computer processor will do two loop executions and provide loop termination conditions based on relating set values (of I or J).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to modify an execution mechanism as suggested by Stoodley in accordance to the structure disclosed by Albert. Doing so would perform multiple iteration paths in a suitable architecture for dealing with more than two iteration paths for covering complex loop structures usually existed in a real computer-programming algorithm for improving loop performance.

As per claim 13: "*The method of claim 12, further comprising the step of determining whether the first loop termination condition is conditional or fixed and determining whether the second loop termination condition is conditional or fixed*" (See Albert: regarding, I=1:10, J=1:10)

As per claim 14: Claim 14 recites method of executing instructions by a processor, the method comprising: executing a first loop including a first instruction within a body of the first loop and a second loop including a first instruction within a body of the second loop, wherein the first instruction of the first loop and the first instruction of the second loop are a same instruction at the same address.

The recitation, including "*first instruction*" in association with "*same instruction at the same address*", has the functionality corresponding to the claim 12 in term of "*loop termination condition*". Therefore claim 14 is rejected in the same reason set forth in connecting to the rejection of claim 12.

As per claim 15: Claim 15 recites a method of executing instructions by a processor, the method comprising: executing a first loop including a last instruction within a body of the first loop and a second loop including a last instruction within a body of the second loop, wherein the last instruction of the first loop and the last instruction of the second loop are the same instruction at the same address.
The recitation, including "*last instruction*" in association with "*same instruction at the same address*", has the functionality corresponding to the claim 12 in term of "*loop termination condition*". Therefore claim 15 is rejected in the same reason set forth in connecting to the rejection of claim 12.

(c) Claims 16-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Albert et al., "Data Parallel Computers and the FORALL Statements" 1990 IEEE.

As per claim 16: Albert discloses a data structure, Forall (.,. ,.,.) (Albert: see page 391, first column, data structure 'Forall') to cover the teaching, "*A processor instruction comprising: a first field that indicates a first termination condition for a first execution loop* (Albert: see page 391, first column, particularly, 'forall' with 'I=1:N'); *and a second field that indicates a second termination condition for a second execution loop* (Albert: see page 391, first column, particularly, 'forall' with 'J=1:M').

Albert does not address the data structure in a manner of a processor instruction. However, it would be an intended use of the Albert's data structure, for the processor instruction because multiple iteration paths are often dealt in any computer-programming algorithm.
Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to intend a similar structure of Albert for a processor instruction. Doing so would resolve multiple iteration paths in computer programming algorithm for dealing with complex loop structures for improving loop performance.
As per claim 17: "*The processor instruction of claim 16, wherein the first execution loop is a same loop as the second execution loop*" (Albert: see page 391, first column, particularly, 'forall' with I and J).
As per claim 18: "*The processor instruction of claim 16, further comprising a third field that indicates a first end-of-loop location*" (Albert: see page 391, first column, particularly, 'forall' with "N').

As per claim 19: "*The processor instruction of claim 18, further comprising a fourth field that indicates a second end-of-loop location*" (Albert: see page 391, first column, particularly, 'forall' with 'M').

Conclusion

4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Kyushima et al., US No. 6,055,627.

Fleck et al., US No. 6,085,315.

Hensley et al., "Active Page Architectures for Media Processing", November 1999.

Warter et al., "Enhanced Modulo Scheduling for Loops with Conditional branches", December 1992.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (703) 308-9049. The examiner can normally be reached on Monday-Friday from 8:00 AM to 5:30 PM ET. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam, can be reached on (703) 305-4552.

The fax phone numbers for this Group are:

Official: (703) 746-7239; After Final: (703) 746-7238; Non-Official: (703) 746-7240.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.

TUAN DAM
SUPERVISORY PATENT EXAMINER

TTV
August 17, 2003